

WHAT IS CLAIMED IS:

1. A system for dynamically configuring a virtual volume associated with a host system, comprising:

a set of storage devices, each of which includes physical block addresses that store data associated with the virtual volume; and

a network switch system connecting the host system and the set of storage devices, and including:

a set of storage processors each maintaining virtual volume objects including at least one of (i) first tier objects reflecting a relationship between the physical block addresses and one or more logical partitions of virtual volume data, and (ii) second tier objects reflecting a logical configuration of the virtual volume,

wherein the network switch system uses the virtual volume objects to dynamically update the virtual volume during runtime of the network switch system.

2. The system of claim 1, wherein the network switch system dynamically updates the virtual volume based on a host system request.

3. The system of claim 1, wherein the network switch system dynamically updates the virtual volume by at least one of adding a virtual volume object to a storage processor, removing a virtual volume object from a storage processor, and moving a virtual volume object from one storage processor to another storage processor.

4. The system of claim 1, wherein storage processors having a first tier object are connected to a storage device storing virtual volume data and storage processors having a second tier object are connected to the host system.

5. The system of claim 1, wherein the network switch system includes a Virtualization Block Manager (VBM) component that, based on a host system request, restructures a logical tree reflecting relationships between the second tier and first tier objects of the virtual volume.

6. The system of claim 5, wherein the network switch system further includes a Virtualization Coherency Manager (VCM) that assigns the first tier objects to selective ones of the storage processors and the second tier objects to selective ones of the second tier storage processors based on the restructured logical tree.

7. The system of claim 5, wherein when the host system request requires the VBM component to add a new second tier object to a target storage processor maintaining a first tier object, the VBM component configures the new second tier object to include a Local Reference Node (LRN) that references the first tier object.

8. The system of claim 7, wherein the VBM component configures the second tier object to include a Remote Reference Node (RRN) that references another first tier object maintained by a storage processor different from the target storage processor.

9. The system of claim 5, wherein when the host system request requires the VBM component to add a new first tier object to a target storage processor maintaining a second tier object, the VBM component configures the second tier object to include a Local Reference Node (LRN) that references the new first tier object.

10. The system of claim 5, wherein when the host system request requires the VBM component to add to a target storage processor a new first tier object that is logically related to a second tier object maintained in a different storage processor, the VBM component configures the second tier object to include a Remote Reference Node (RRN) that references the new first tier object.

11. The system of claim 6, wherein when the host system request requires the VBM component to remove an existing second tier object tree from a target storage processor, the VCM deletes all second tier objects in the second tier tree before deleting any first tier objects that are solely referenced by the removed second tier object tree.

12. The system of claim 11, wherein the target storage processor maintains an existing first tier object referenced by the existing second tier object tree and by a remote second tier object maintained by a remote storage processor, and wherein the VCM maintains the existing first tier object when deleting the existing second tier object.

13. The system of claim 6, wherein when the host system request requires the VBM component to remove an existing first tier object from a target storage processor, the VCM deletes all references to the existing first tier object from any second tier objects.

14. The system of claim 6, wherein when the host system request requires the VBM component to move an existing first tier object from a first storage processor to a second storage processor having a remote reference to the existing first tier object, the VCM sends a new second tier object tree to the first and second storage processors that removes any references to the existing first tier object.

15. The system of claim 14, wherein the VCM sends a new first tier object to the second storage processor that deletes the remote reference to the existing first tier object from the second storage processor.

16. The system of claim 15, wherein the VCM sends a copy of the existing first tier object to the second processor following deletion of the remote reference.

17. The system of claim 16, wherein the VCM sends a new second tier object tree to the second storage processor having a new local reference to the copy of the existing first tier object.

18. The system of claim 17, wherein the VCM sends the new second tier object tree to a third storage processor with a remote reference to the copy of the existing first tier object sent to the second storage processor.

19. The system of claim 2, wherein the network switch system dynamically updates the virtual volume by collecting state information from the storage processors reflecting a current view of the virtual volume and reconfiguring a logical tree reflecting a logical relationship between the virtual volume objects based on the state information and the host system request.

20. The system of claim 19, wherein the current view of the virtual volume includes information reflecting which storage processors maintain first tier objects and which storage processors maintain second tier objects.

21. The system of claim 1, wherein each storage processor includes a virtualization state manager (VSM) that is configured to manage a local version of the virtual volume.

22. The system of claim 22, wherein each storage processor VSM is configured to manage any of the virtual volume objects maintained by the respective storage processor.

23. The system of claim 6, wherein a single storage processor includes a Master VSM (MVSM) that is in an active state.
24. The system of claim 23, wherein the MVSM is configured to determine which virtual volume objects are affected by the restructured logical tree.
25. The system of claim 24, wherein the VCM assigns the first and second tier objects to respective first and second tier storage processors based on the affected virtual volume objects determined by the MVSM.
26. The system of claim 23, wherein the network switch system designates the single storage processor as a Master Virtualization Storage Processor (MVSP) by activating the MVSM in the designated MVSP.
27. The system of claim 26, wherein non-MVSP storage processors include an MVSM that in an inactive state when the single storage processor is designated as the MVSP.
28. The system of claim 2, wherein the network switch system includes a Virtualization Coherency Manager (VCM) that updates virtual volume assignments to the storage processors based on the host system request.

29. The system of claim 28, wherein the network switch system includes a Virtualization Block Manager (VBM) that creates the first and second tier objects based on a user request to update the virtual volume.

30. A method for dynamically updating a virtual volume in a multi-tier virtualization storage system including a set of storage devices storing virtual volume data and connected to a first set of storage processors, and a second set of storage processors connected to a host system associated with the virtual volume, wherein the virtual volume is defined by a set of virtual volume objects associated with selected ones of the first and second set storage processors, the method comprising:

receiving a request from the host system to adjust the virtual volume data;

determining which virtual volume objects are affected by the request to adjust the virtual volume data;

updating the virtual volume based on the affected virtual volume objects and host system request; and

allowing the host system to access the updated virtual volume.

31. The method of claim 30, wherein updating the virtual volume includes:

updating a logical tree reflecting relationships between the virtual volume objects based on the request to adjust the virtual volume data.

32. The method of claim 31, wherein the virtual volume objects include first tier and second tier objects and wherein updating the virtual volume further includes:

assigning first tier objects to selective ones of the first set of storage processors and second tier objects to selective ones of the second set of storage processors based on the updated logical tree.

33. The method of claim 31, wherein updating the logical tree includes:
updating a definition for at least one of the virtual volume objects based on the
host system request.
34. The method of claim 32, wherein the first tier objects include references to
virtual volume data partitioned in the storage devices.
35. The method of claim 32, wherein the second tier objects include
information reflecting a logical configuration of the virtual volume.
36. The method of claim 35, wherein the second tier objects includes
references to selective ones of the first tier objects based on the logical configuration.
37. The method of claim 30, wherein updating the virtual volume includes at
least one of adding a virtual volume object to a storage processor, removing a virtual
volume object from a storage processor, and moving a virtual volume object from one
storage processor to another storage processor.
38. The method of claim 30, wherein the virtual volume objects include first
tier and second tier objects and updating the virtual volume includes restructuring a
logical tree reflecting relationships between the second tier and first tier objects of the
virtual volume.

39. The method of claim 37, wherein the virtual volume objects include first tier and second tier objects and wherein adding a virtual volume object includes:

adding a new second tier object to a target storage processor maintaining a first tier object.

40. The method of claim 39, wherein adding the new second tier object includes:

configuring the new second tier object to include a Local Reference Node (LRN) that references the first tier object.

41. The method of claim 40, wherein configuring the new second tier object includes configuring the new second tier object with a Remote Reference Node (RRN) that references another first tier object maintained by a storage processor different from the target storage processor.

42. The method of claim 37, wherein the virtual volume objects include first tier and second tier objects and wherein adding a virtual volume object includes:

adding a new first tier object to a target storage processor maintaining a second tier object.

43. The method of claim 42, wherein adding the new first tier object includes:

configuring the second tier object to include a Local Reference Node (LRN) that references the new first tier object.

44. The method of claim 37, wherein the virtual volume objects include first tier and second tier objects and wherein adding a virtual volume object includes:

adding to a target storage processor a new first tier object that is logically related to a second tier object maintained in a different storage processor.

45. The method of claim 44, wherein adding the new first tier object includes:

configuring the second tier object to include a Remote Reference Node (RRN) that references the new first tier object.

46. The method of claim 37, wherein the virtual volume objects include first tier and second tier objects and at least one existing second tier object tree including some of the second tier objects, and wherein removing a virtual volume object includes:

removing the existing second tier object tree from a target storage processor.

47. The method of claim 46, wherein removing the existing second tier object tree includes:

deleting all second tier objects in the second tier tree before deleting any first tier objects that are solely referenced by the removed second tier object tree.

48. The method of claim 47, wherein the target storage processor maintains an existing first tier object referenced by the existing second tier object tree and by a

remote second tier object maintained by a remote storage processor, and wherein removing the existing second tier object tree includes:

maintaining the existing first tier object when deleting the second tier objects from the existing second tier object tree.

49. The method of claim 37, wherein the virtual volume objects include first tier and second tier objects and at least one existing second tier object tree including some of the second tier objects, and wherein removing a virtual volume object includes:

removing an existing first tier object from a target storage processor.

50. The method of claim 49, wherein removing the existing first tier object includes:

deleting all references to the existing first tier object from any second tier objects.

51. The method of claim 37, wherein the virtual volume objects include first tier and second tier objects and at least one existing second tier object tree including some of the second tier objects, and wherein moving a virtual volume object includes:

moving an existing first tier object from a first storage processor to a second storage processor having a remote reference to the existing first tier object.

52. The method of claim 51, wherein moving the existing first tier object includes:

sending a new second tier object tree to the first and second storage processors that removes any references to the existing first tier object.

53. The method of claim 52, wherein moving the existing first tier object includes:

sending a new first tier object to the second storage processor that deletes the remote reference to the existing first tier object from the second storage processor.

54. The method of claim 53, further including:

sending a copy of the existing first tier object to the second processor following deletion of the remote reference.

55. The method of claim 54, further including:

sending another second tier object tree to the second storage processor having a new local reference to the copy of the existing first tier object.

56. The method of claim 55, further including:

sending the another second tier object tree to a third storage processor with a remote reference to the copy of the existing first tier object sent to the second storage processor.

57. The method of claim 31, wherein determining includes:

sending the updated logical tree to a designated storage processor and receiving information from the designated storage processor indicating which virtual volume objects are affected by the request to adjust the virtual volume data. .

58. The method of claim 57, further comprising:

reassigning the virtual volume objects to selected ones of the first and second set storage processors based on the information received from the designated storage processor.

59. A computer-readable medium including instructions for performing a method, when executed by a processor, for dynamically updating a virtual volume in a multi-tier virtualization storage system including a set of storage devices storing virtual volume data and connected to a first set of storage processors, and a second set of storage processors connected to a host system associated with the virtual volume, wherein the virtual volume is defined by a set of virtual volume objects associated with selected ones of the first and second set storage processors, the method comprising:

receiving a request from the host system to adjust the virtual volume data;

determining which virtual volume objects are affected by the request to adjust the virtual volume data;

updating the virtual volume based on the affected virtual volume objects and host system request; and

allowing the host system to access the updated virtual volume.

60. A system for dynamically updating a virtual volume in a multi-tier virtualization storage environment including a set of storage devices storing virtual volume data and connected to a first set of storage processors, and a second set of storage processors connected to a host system associated with the virtual volume, wherein the virtual volume is defined by a set of virtual volume objects associated with selected ones of the first and second set storage processors, the system comprising:

- means for receiving a request from the host system to adjust the virtual volume;
- means for determining which virtual volume objects are affected by the request to adjust the virtual volume data;
- means for updating the virtual volume based on the affected virtual volume objects and host system request; and
- means for allowing the host system to access the updated virtual volume.